## Day 1 Tutorials

If you are using a Windows OS, be sure to save all files to your external USB drive because we will be booting into Linux Ubuntu later and you will want to access the files that are on your USB drive.

Start by making a directory on your USB drive called phyloWorkshop. Create another directory inside phyloWorkshop called day1Practical. You can do all of that on one line on the terminal if you know the name of your USB drive. Substitute the name of your USB drive in the appropriate places below.

```
mkdir /media/ubuntu/NameOfUSBDrive/phyloWorkshop && mkdir /media/ubuntu/NameOfUSBDrive/phyloWorkshop/day1Practical
```

Save all files you create today in day1Practical.

# Acquiring Data from GenBank

We are going to use publicly available resources to retrieve sequence data for comparison to sequence data we have collected for which we have a hypothesis about its origin (i.e. genus).

The first thing we are going to do is to make sure that we have sequenced the correct organism by comparing it to sequences that have been deposited in GenBank, which is hosted by the National Center for Biotechnology Information (NCBI).

What is GenBank?

`What is GenBank? GenBank ® is the NIH genetic sequence database, an annotated collection of all publicly available DNA sequences (Nucleic Acids Research, 2013 Jan;41(D1):D36-42). GenBank is part of the International Nucleotide Sequence Database Collaboration, which comprises the DNA DataBank of Japan (DDBJ), the European Nucleotide Archive (ENA), and GenBank at NCBI. These three organizations exchange data on a daily basis. More info: https://www.ncbi.nlm.nih.gov/genbank/`

Using BLAST to query Genbank

In order to search the GenBank database, we need to have some means of comparing our sequence to the sequences that are most similiar.

What is BLAST?

It is an algorithm for comparing sequences.

```
The Basic Local Alignment Search Tool (BLAST) finds regions of local similarity between
sequences. The program compares nucleotide or protein sequences to sequence databases and
calculates the statistical significance of matches. BLAST can be used to infer functional and
evolutionary relationships between sequences as well as help identify members of gene
families.
```

We amplified the nuclear ribosomal internal transcribed spacer region and sequenced it. Here is the sequence:

```
> mystery_organism_nrITS
acaaggtctccgtaggtgaacctgcggagggatcattacacaatacaatatgaaggctgtccgcagctggagtattttattacccttgtctttt
gcgcacttgttgtttcctgggcgggttcgctcgccaccaggaccaccaaataaaccttttttatgcagttgcaatcagcgtcagtacaaacaat
gtaaatcatttacaactttcaacaacggatctcttggttctggcatcgatgaagaacgcagcgaaatgcgatacgtagtgtgaattgcagaatt
cagtgaatcatcgaatctttgaacgcacattgcgccctttggtattccaaagggcatgcctgttcgagcgtcatttgtaccctcaagctttgct
tggtgttgggcgtttttgtctttggtcgcccaaagactcgccttaaagtgattggcagccggcctttctggtttcgcagcgcagcacatttttg
cgcttgccatcagcaaaacggcaatccatcaagcctccttctcacgtttgacctcggatcaggtagggatacccgctgaacttaagcatatc
```

- Open a browser to the BLAST website: https://blast.ncbi.nlm.nih.gov/Blast.cgi
- Open a separate browser to https://digitalworldbiology.com/blast
- Resize your browser so that you can see both the tutorial and the BLAST page side by side.

We will walk through the BLAST tutorial linked above using the "mystery_organism_nrITS" as a query sequence to gain some familiarity with how to use BLAST and interpret the output. We will also determine if we sequenced what we thought it was.

- Do a BLAST search with the "mystery_organism_nrITS" using both the megablast and blastn algorithms.
  - Do you see any differences between the two?
  - Read here and/or here to learn more about the different alogrithms.

Answer the following questions:

1. What is the length of the sequence?
2. How many genera and species are represented in the top 10 matches?
3. What percentage of our query sequence aligns with the top match?
4. What percentage of nucleotides are identical to the top match?
5. How many of the top 10 matches are not identical over the length of the query sequence?
6. What is the probability that you would find a match the same as the top match in a database of random sequences the size of the current GenBank database given the

length of the query sequence?

7. What nucleotide position in the query sequence is the mismatch with Curvularia chiangmaiensis strain CPC 28829?
8. From which host was Curvularia chiangmaiensis strain CPC 28829 collected?
9. Is this sequence published? Where?
10. Is the top match published? Where?

Query GenBank with the two sequences below and answer the following questions:

1. What region of the genome does this sequence represent?
2. Is this a coding or non-coding sequence?
3. How many species are represented in the top 15 matches? Do they all have the same sequence identity with the query sequence?
4. Are the top two matches published?
5. What is the GenBank accession number for the top match?

`> mystery_organism_locus2`

GACTGCGCCATTCTCAAATCATTGCCGCCGGTACTGGTGAGTTCGAGGCTGGTATCTCCAAG
GATGGTCAGACTCGTGAGCACGCTCTGCTCGCCTACACCCTCGGTGTCAAGCAGCTCATCG
TCGCCATCAACAAGATGGACACCACCAAGTGGTCTGAGGAGCGTTACCAGGAAATCATCAAG
GAGACCTCCAACTTCATCAAGAAGGTCGGCTACAACCCCAAGCACGTTCCCTTCGTCCCCAT
CTCCGGTTTCAACGGAGACAACATGATTGAGGCTTCCACCAACTGCCCCTGGTACAAGGGTT
GGGAGAAGGAGACCAAGGCCAAGGCCACTGGTAAGACCCTCCTCGAGGCCATCGACGCCA
TCGACCCCCCTGTCCGTCCTACCGACAAGCCCCTCCGCCTTCCCCTCCAGGATGTCTACAA
GATTGGTGGTATTGGCACGGTCCCCGTCGGTCGTGTCGAGACCGGTATCATCAAGCCCGGT
ATGGTCGTCACCTTCGCCCCCGCTGGTGTCACCACTGAAGTCAAGTCCGTCGAGATGCACC
ACGAGCAGCTCACCGAGGGTGTCCCCGGTGACAACGTCGGCTTCAACGTCAAGAACGTCTC
CGTCAAGGAGATCCGTCGTGGTAACGTTGCCGGTGACTCCAAGAACGACCCCCCCAAGGGT
TGCGAGTCCTTCAACGCCCAGGTCATCGTCCTCAACCACCCCGGTCAGGTCGGTGCCGGTT
ACGCACCAGTCCTTGACTGCCACACTGCCCACATTGCTTGCAAGTTCTCCGAGCTCCTCGAG
AAGATCGACCGCCGTACCGGAAAGTCTGTTGAGAACTCCCCCAAGTTCATCAAGTCCGGTGA
CGCTGCCATCGTCAAGATGGTTCCCTCCAAGCCCATGTGCGTTGAGGCTTTCACTGACTACC
CTCCTCTCGGTCGTTTCGCCGTCCGTGACATGCGTCAGACGT``

`> mystery_organism_locus3`

TCAACGGCTTTCGGTCGCATTGGCCGTATCGTCTTCCGCAATGCGTAGGTGCCCTTGAATCC
ATTGATTCAGCGTGTATCGAAGCTAATCGAAGCTCGCAGCATCGAGCACAACGACGTCGAGA
TTGTCGCCGTGAACGACCCCTTCATCGAGCCCCACTACGCTGTAAGCATCCCCAGCACAGA
ATCCTTCCGTCAGAGCGATGCTTTGCATCATTGATTCCATCCTGGCATGATCCATTGGCGGAA
CAGTACAAGCTAACATGTCCATAGGCATACATGCTCAAGTATGACAGCACACACGGCCAGTT
CAAGGGCGACATCAAGGTTGACGGCAACAACCTGACTGTCAACGGCAAGACCGTCCGCTTC
CACATGGAGAAGGACCCCGCCAACATCCCATGGAGCGAGACCGGCGCTTACTACGTTGTTG
AGTCCACTGGTGTCTTCACCACCACCGAGAAGGCCAAGGCTCACTTGAAGGGTGGAGCCAA

GAAGGTTGTCATCTCTGCTCCCTCCGCCGATGCCCCTATGTTCGTCATGGGTGTCAACCACGAGACCTACAAGTCTGACATTGAGGTCCTCTCCAAC

Using publicly available sequence data from GenBank

Based on the BLAST searches above, it seems that our sequence represents a species of *Curvularia*. The BLAST results are not conclusive and a similarity based search (BLAST) is not the best means to determine what species we have. We need to compare it to published sequences from representative isolates of closely related species, preferably sequences from type or ex-type specimens.

A little bit of background on our isolate. It was collected from a leaf spot on *Zea mays* in Louisiana. We need to determine if this is a species that has been previously associated with this host in Louisiana or the United States. The first thing we need to do is identify the species.

The first place to look for sequence data is to look for recent comprehensive phylogenetic studies of the genus *Curvularia*. There are a few recent papers that include several species in the genus for which they have sequenced the same loci as those represented by the sequences above. The authors of these papers have submitted their data to GenBank. We will acquire the data from GenBank so that we may run phylogenetic analyses later on these data.

GenBank - Online Interface

The sequences below are from two papers:

1. `Marin-Felix Y, Senwanna C, Cheewangkoon R, Crous PW. 2017. New species and records of Bipolaris and Curvularia from Thailand. Mycosphere 8:1556–1574. doi10.5943/mycosphere/8/9/11`

2. `Manamgoda, Rossman, Castlebury, Chukeatirote, and Hyde. 2015. A taxonomic and phylogenetic re-appraisal of the genus Curvularia (Pleosporaceae): human and plant pathogens. Phytotaxa 212 (3): 175–198`

Here are the GenBank accession numbers from these papers:

1. `MF490804 MF490805 MF490806 MF490807 MF490808 MF490809 MF490810 MF490811 MF490812 MF490813 MF490814 MF490815 MF490817 MF490818 MF490816 MF490819 MF490822 MF490820 MF490821 MF490823 MF490825 MF490824`

2. `AF071325 HE861850 KJ909780 KJ909782 JX256420 JX256421 HE861834 HE861832 HE861833 JX256424 KP400631 JX256425 KP400633 JN601026 AF081448 KJ415542 KP400632 JX256423 JX256439 JX256441 KP400630 KP400634 KJ922372 KJ909765 HG778984 HG779021 JN192373 KJ415544 KJ415545 JN192375 JX256426 KJ909781 JN192376 KJ415546 JX256427 JN601028 JN192377 KJ415547 KJ415548 HE861848 KP400635 KP400636 KP400637 KP400638 KJ415543 KP400639 JX256436 KP400640 KP400641 KP400642 HE861837 KJ922375 KJ922374 JX256428 KP400646 KJ909770 HG779002 JN192381 KP400643 JX256429 KP400644 JX256430 KP400645 KP400647 KP400648 KP400649 HE861836 JN192379 KJ415550 AF081449 KJ909772 JN601033 KP400650 JN192384 KJ909774 KJ922380 JN192385 KJ922373 KJ922376 HE861838 HE861842 JN192386 KJ415555 KJ909783 KJ909766 EF175940 KJ415558 JN192387 KJ909777 KC424596 KP400651 KM230395 JX256445 JX256443 KP400655 HE861826 JX256444 KP400656 JN192388 KJ415559 JX256433 HG779024 KP400652 JX256437 KP400653 JX256442 KP400654 KJ922377`

Open a browser and paste the following url: https://www.ncbi.nlm.nih.gov/genbank/

You will see a search bar with a dropdown menu to the left of the search bar. Look at the dropdown menu to see all of the databases that are available for searching. We will search the 'Nucleotide' database for our sequences.

Start by pasting the first three accession numbers into the search bar: `MF490804 MF490805 MF490806`

We will explore some of the features of the search results together, including the GenBank (gb) and fasta file formats, population datasets (popset), and the taxonomy database. Finally, we will see how to export these file formats to a file on our computer.

Once you have some familiarity with the GenBank database, download the sequences above into two separate files for each set of sequences; one in GenBank format and the other in fasta format. Name the files `CurvulariaMarinFelix.gb`, `CurvulariaMarinFelix.fasta`, `CurvulariaManamgoda.gb`, and `CurvulariaManamgoda.fasta`, respectively. Make sure that you do not put spaces or any other special characters in the filenames. Save them to your USB drive.

We will return to the files above, but let us think again about the question we are trying to answer. We want to know if the isolate that we collected is the first of its kind from corn (*Zea mays*) and from Louisiana or the United States. We certainly need to search the literature to see if anything has been published from Zea in Louisiana or the United States, but it would also be worthwhile to search GenBank for any sequences of Curvularia from Louisiana or the United States on corn.

Try the following search to limit your search to sequences of *Curvularia* from *Zea* in the USA: `Curvularia[organism] AND USA[country] AND Zea[host]`

- How many sequences does this query return?
- How many different genes are represented?

Save the nrITS from published papers into a two new files: `CurvulariaUSZea.gb` and `CurvulariaUSZea.fasta`

You can find more information about search fields here and the available search fields and qualifiers here.

You can also use the 'Advanced' search to build a custom search.

Using Entrez Efetch Utilities to retrieve sequences

Now we are going to repeat the exercise above using E-utilities, the application programming interface (API) for GenBank. You can read more about it here.

If you are using a Windows PC, you are going to boot your computer into the Linux Ubuntu operating system and install e-utilities. This will take a few minutes.

If you are using a UNIX-based operating system (Mac or Linux) you should be able to install e-utilities immediately.

Go to https://github.com/phylodojo/files/blob/master/install.efetch.txt to see the commands that you need to paste into your terminal to install E-Utilities.

Check that E-Utilities is installed by typing `esearch` into the terminal. It should return the following if it is installed properly: "Must supply -db database on command line"

** Retrieve sequences from GenBank in fasta format **

The format for searching the nucleotide database is as follows:

```
esearch -db nucleotide -query "Accession Numbers Here" | efetch -format fasta
```

Try it out:

```
esearch -db nucleotide -query "MF490804 MF490805 MF490806" | efetch -format fasta
```

Now to redirect the output from the standard out to a file with >>

```
esearch -db nucleotide -query "MF490804 MF490805 MF490806" | efetch -format fasta >> test.fasta
```

Try it out by retrieving sequences in GenBank format:

```
esearch -db nucleotide -query "MF490804 MF490805 MF490806" | efetch -format gb
```

Now to redirect the output from the standard out to a file with >>

```
esearch -db nucleotide -query "MF490804 MF490805 MF490806" | efetch -format gb >> test.gb
```

- Now retrieve the same accession numbers in both fasta and GenBank format as you did using the online interface. Save these files with same filenames, except add .esearch to the files.

Building your own local BLAST database

Sometimes you will need to retrieve individual locus data from whole genome sequences that are publically available. To do this, you can use a homologous sequence to retrieve data from the genome by setting up your own local BLAST database.

Before moving on, you will need to install ncbi-blast tools. You can do this with the install script that I wrote here: https://github.com/phylodojo/files/blob/master/

- Click on "install.phylo.day1.sh" > right-click on "RAW" > Save as "install.phylo.day1.sh" onto your Desktop
- open a terminal and type: `cd home/ubuntu/Desktop`
- `chmod +x install.phylo.day1.sh`
- `./install.phylo.day1.sh`

This should install ncbi-blast and a few other scripts we will use later. It will take awhile to run. If it is finished in less than 10 seconds, please let me know so we can try to address the problem.

To make sure that ncbi-blast installed properly:
`which blastn`

This should return "/usr/bin/blastn"

Setup BLAST database

We are going to retrieve the genome sequence for *Aspergillus flavus* from GenBank.

- `cd /home/ubuntu`
- `mkdir blast`
- `export BLASTDB=/home/ubuntu/blast`
- `cd /home/ubuntu/blast`
- `wget ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/006/275/GCF_000006275.2_JCVI-afl1-v2.0/GCF_000006275.2_JCVI-afl1-v2.0_genomic.fna.gz`
- `gunzip GCF_000006275.2_JCVI-afl1-v2.0/GCF_000006275.2_JCVI-afl1-v2.0_genomic.fna.gz`

Check to make sure you have the fasta file unzipped:

```
head GCF_000006275.2_JCVI-afl1-v2.0/GCF_000006275.2_JCVI-afl1-v2.0_genomic.fna
```

This will show you the first 10 lines of the fasta file.

** Build the BLAST database **

```
makeblastdb -in GCF_000006275.2_JCVI-afl1-v2.0/GCF_000006275.2_JCVI-afl1-v2.0_genomic.fna -out flavusDB -dbtype nucl
```

- Grab accession number KX100865 in fasta format using esearch and efetch into Clunata.tef.fasta
- Use blastn to see if querying the *Aspergillus flavus* genome with the *Curvularia lunata* translation elongation factor sequence returns a match: `blastn -db flavusDB -query Clunata.tef.fasta -outfmt '6 sseqid sseq'`

The outfmt specifiers above (sseqid sseq) return the sequence identifier and the aligned portion of the sequences in the database that match the query sequence. You will notice that this is close to fasta format, but not quite. We can pass the results of our BLAST search to some unix commands to save the output in fasta format to a file.

```
blastn -db flavusDB -query Clunata.tef.fasta -outfmt '6 sseqid sseq' | awk 'BEGIN{FS="\t"; OFS="\n"}{gsub(/-/, "", $2); print "> Aflavus.tef."$1,$2}'
```

There are several blast tools that are installed with ncbi-blast+, including blastn and blastp. In order to see all of the commands that are available: `blastn -h`
To see detailed descriptions of each command and the options: `blastn -help`
You can also see the options and outfmt specifiers here.

If time permits
GenbankProcess.py
reformatFasta.py

If there is time to work with the above, we will need to run python and dendropy install commands that have been commented out in install.phylo.day1.sh